Adafruitframebuf Library Documentation

Release 1.0

Kattni Rembor

Apr 07, 2020

Contents

CircuitPython frambuf module, based on the Python frambuf module.

CHAPTER 1

Dependencies

This driver depends on:

- Adafruit CircuitPython

Please ensure all dependencies are available on the CircuitPython filesystem. This is easily achieved by downloading the Adafruit library and driver bundle.

1.1 Installing from PyPI

On supported GNU/Linux systems like the Raspberry Pi, you can install the driver locally from PyPI. To install for current user:

```
pip3 install adafruit-circuitpython-framebuf
```

To install system-wide (this may be required in some cases):

```
sudo pip3 install adafruit-circuitpython-framebuf
```

To install in a virtual environment in your current project:

```
mkdir project-name && cd project-name
python3 -m venv .env
source .env/bin/activate
pip3 install adafruit-circuitpython-framebuf
```

CHAPTER 2

Usage Example

See example in /examples/framebuf_simpletest.py

CHAPTER 3

Contributing

Contributions are welcome! Please read our Code of Conduct before contributing to help this project stay welcoming.

CHAPTER 4

Documentation

For information on building library documentation, please check out this guide.

CHAPTER 5

Table of Contents

5.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/framebuf_simpletest.py

```
import adafruit_framebuf

print("framebuf test will draw to the REPL")

WIDTH = 32
HEIGHT = 8

buffer = bytearray(round(WIDTH * HEIGHT / 8))
fb = adafruit_framebuf.FrameBuffer(
    buffer, WIDTH, HEIGHT, buf_format=adafruit_framebuf.MVLSB
)

# Ascii printer for very small framebufs!
def print_buffer(the_fb):
    print("." * (the_fb.width + 2))
    for y in range(the_fb.height):
        print(".", end="")
        for x in range(the_fb.width):
            if fb.pixel(x, y):
                print("*", end="")
            else:
                print(" ", end="")
        print(".")
    print("." * (the_fb.width + 2))


# Small function to clear the buffer
```

(continues on next page)

(continued from previous page)

```
def clear_buffer():
    for i, _ in enumerate(buffer):
        buffer[i] = 0


print("Shapes test: ")
fb.pixel(3, 5, True)
fb.rect(0, 0, fb.width, fb.height, True)
fb.line(1, 1, fb.width - 2, fb.height - 2, True)
fb.fill_rect(25, 2, 2, 2, True)
print_buffer(fb)

print("Text test: ")
# empty
fb.fill_rect(0, 0, WIDTH, HEIGHT, False)

# write some text
fb.text("hello", 0, 0, True)
print_buffer(fb)
clear_buffer()

# write some larger text
fb.text("hello", 8, 0, True, size=2)
print_buffer(fb)
```

5.2 adafruit_framebuf

CircuitPython pure-python framebuf module, based on the micropython framebuf module.

- Author(s): Kattni Rembor, Tony DiCola, original file created by Damien P. George

5.2.1 Implementation Notes

Hardware:

- Adafruit SSD1306 OLED displays

Software and Dependencies:

- Adafruit CircuitPython firmware for the supported boards: https://github.com/adafruit/circuitpython/releases

class adafruit_framebuf.**BitmapFont**(*font_name='font5x8.bin'*)
A helper class to read binary font tiles and 'seek' through them as a file to display in a framebuffer. We use file access so we dont waste 1KB of RAM on a font!

deinit()
Close the font file as cleanup.

draw_char(*char*, *x*, *y*, *framebuffer*, *color*, *size=1*)
Draw one character at position (x,y) to a framebuffer in a given color

width(*text*)
Return the pixel width of the specified text message.

class adafruit_framebuf.**FrameBuffer**(*buf*, *width*, *height*, *buf_format=0*, *stride=None*)
FrameBuffer object.

Parameters

- **buf** – An object with a buffer protocol which must be large enough to contain every pixel defined by the width, height and format of the FrameBuffer.
- **width** – The width of the FrameBuffer in pixel
- **height** – The height of the FrameBuffer in pixel
- **buf_format** – Specifies the type of pixel used in the FrameBuffer; permissible values are listed under Constants below. These set the number of bits used to encode a color value and the layout of these bits in `buf`. Where a color value c is passed to a method, c is a small integer with an encoding that is dependent on the format of the FrameBuffer.
- **stride** – The number of pixels between each horizontal line of pixels in the FrameBuffer. This defaults to `width` but may need adjustments when implementing a FrameBuffer within another larger FrameBuffer or screen. The `buf` size must accommodate an increased step size.

blit()
blit is not yet implemented

circle(*center_x*, *center_y*, *radius*, *color*)
Draw a circle at the given midpoint location, radius and color. The `circle` method draws only a 1 pixel outline.

fill(*color*)
Fill the entire FrameBuffer with the specified color.

fill_rect(*x*, *y*, *width*, *height*, *color*)
Draw a rectangle at the given location, size and color. The `fill_rect` method draws both the outline and interior.

hline(*x*, *y*, *width*, *color*)
Draw a horizontal line up to a given length.

image(*img*)
Set buffer to value of Python Imaging Library image. The image should be in 1 bit mode and a size equal to the display size.

line(*x_0*, *y_0*, *x_1*, *y_1*, *color*)
Bresenham's line algorithm

pixel(*x*, *y*, *color=None*)
If `color` is not given, get the color value of the specified pixel. If `color` is given, set the specified pixel to the given color.

rect(*x*, *y*, *width*, *height*, *color*, *, *fill=False*)
Draw a rectangle at the given location, size and color. The `rect` method draws only a 1 pixel outline.

rotation
The rotation setting of the display, can be one of (0, 1, 2, 3)

scroll(*delta_x*, *delta_y*)
shifts framebuf in x and y direction

text(*string*, *x*, *y*, *color*, *, *font_name='font5x8.bin'*, *size=1*)
Place text on the screen in variables sizes. Breaks on to next line.

Does not break on line going off screen.

vline(*x*, *y*, *height*, *color*)
Draw a vertical line up to a given length.

class adafruit_framebuf.**FrameBuffer1**(*buf*, *width*, *height*, *buf_format=0*, *stride=None*)
FrameBuffer1 object. Inherits from FrameBuffer.

class adafruit_framebuf.**MHMSBFormat**

static fill(*framebuf*, *color*)
completely fill/clear the buffer with a color

static fill_rect(*framebuf*, *x*, *y*, *width*, *height*, *color*)
Draw a rectangle at the given location, size and color. The `fill_rect` method draws both the outline and interior.

static get_pixel(*framebuf*, *x*, *y*)
Get the color of a given pixel

static set_pixel(*framebuf*, *x*, *y*, *color*)
Set a given pixel to a color.

class adafruit_framebuf.**MVLSBFormat**

static fill(*framebuf*, *color*)
completely fill/clear the buffer with a color

static fill_rect(*framebuf*, *x*, *y*, *width*, *height*, *color*)
Draw a rectangle at the given location, size and color. The `fill_rect` method draws both the outline and interior.

static get_pixel(*framebuf*, *x*, *y*)
Get the color of a given pixel

static set_pixel(*framebuf*, *x*, *y*, *color*)
Set a given pixel to a color.

CHAPTER 6

Indices and tables

- genindex
- modindex
- search

Python Module Index

a

Index